

Kube-Apiserver Pod의 지속적인 재시작 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[kube-apiserver란 무엇입니까?](#)

[문제](#)

[근본 원인 분석](#)

[복구 단계](#)

[사후 확인](#)

소개

이 문서에서는 Kube-apiserver Pod의 지속적인 재시작을 복구하는 솔루션에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 다음 주제에 대해 숙지할 것을 권장합니다.

- 도커 및 쿠버네티스
- Cisco SMI(Subscriber Microservices Infrastructure) Ultra Cloud Core CEE(Common Execution Environment)

사용되는 구성 요소

이 문서의 정보는 Kubernetes v1.21.0 버전을 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

kube-apiserver란 무엇입니까?

- Kubernetes Application Programming Interface(API) 서버는 포드, 서비스, 복제 컨트롤러 등을 포함하는 API 객체에 대한 데이터를 검증하고 구성합니다. API Server Services REST(Presentational State Transfer) 작업은 다른 모든 구성 요소가 상호 작용하는 클러스터의 공유 상태에 대한 프론트엔드를 제공합니다.
- Kubernetes API 서버는 요청을 인증 및 검증하고, etcd 데이터 저장소에서 데이터를 검색 및 업

데이트하는 작업을 담당합니다. 실제로, kbe-API 서버는 etcd 데이터 저장소와 직접 상호 작용하는 유일한 구성 요소입니다.

- 다음은 Pod가 클러스터에 생성될 때 kube-API 서버가 수행하는 단계입니다.

a. 사용자 인증

b. 요청 검증

c. 데이터 검색

d. ETCD 업데이트

e. 스케줄러

f. 쿠벨레

- 스케줄러, kube-controller-manager 및 kubelet과 같은 다른 구성 요소는 API 서버를 사용하여 해당 영역에서 클러스터에서 업데이트를 수행합니다.

문제

kube-apiserver-smf-data-master-3 재시작은 지속적으로 관찰됩니다. 이 경우 kubectl CLI `kubectl get pods -A -o wide`를 실행합니다. | `grep apiserver`를 통해 문제를 식별합니다.

```
cloud-user@smf-data-master-1:~$ kubectl get pods -A -o wide | grep apiserver
```

kube-system	kube-apiserver-smf-data-master-1	1/1	Running	4
68d	10.192.1.22	smf-data-master-1	<none>	<none>
kube-system	kube-apiserver-smf-data-master-2	1/1	Running	4
68d	10.192.1.23	smf-data-master-2	<none>	<none>
kube-system	kube-apiserver-smf-data-master-3	0/1	Running	2
68d	10.192.1.24	smf-data-master-3	<none>	<none>

```
cloud-user@smf-data-master-1:~$
```

이러한 오류는 `kubectl 로그 <kube-apiserver_pod_name> -n kube-system`에서 관찰되었습니다.

```
cloud-user@smf-data-master-1:~$ kubectl logs kube-apiserver-smf-data-master-3 -n kube-system
E1116 20:09:52.635602      1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:53.691253      1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:54.751145      1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:55.808782      1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:56.865492      1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:57.906426      1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
```


From Master-3:

```
cloud-user@smf-data-master-3:~$ cat /data/kubernetes/secrets.conf
```

```
apiVersion: apiserver.config.k8s.io/v1
```

```
kind: EncryptionConfiguration
```

```
resources:
```

```
- resources:
```

```
- secrets
```

```
providers:
```

```
- aescbc:
```

```
  keys:
```

```
    - name: key1
```

```
      secret: XK+7mbh3YEnMdqswtySQ1d6QRehg+K6/J1d2e3EnMvI= <<<<<<<<
```

```
- identity: {}
```

```
cloud-user@smf-data-master-3:~$
```

복구 단계

1. 복구의 일부로 master-3의 현재 암호를 백업 파일에 복사합니다.

```
cloud-user@smf-data-master-3:~$ sudo cp /data/kubernetes/secrets.conf  
/data/kubernetes/secrets.conf-bkp
```

2. 비밀번호를 수정하고 Master-3에서 구성한 다음 다른 마스터 노드에 표시된 것과 동일한 값으로 비밀의 값을 변경합니다.

```
cloud-user@smf-data-master-3:~$ sudo vim /data/kubernetes/secrets.conf
```

```
apiVersion: apiserver.config.k8s.io/v1
```

```
kind: EncryptionConfiguration
```

```
resources:
```

```
- resources:
```

```
- secrets
```

```
providers:
```

```
- aescbc:
```

```
keys:
```

```
- name: key1
```

```
secret: XK+7mbh3YEnMdqswtySQ1d6QRehg+K6/J1d2e3EnMvI= <---- Change this value to  
"BG5hleucj1D5ZDkFYUxoGLHHhBA/AeoNruHM0i70/ZI=" as in other Master nodes
```

```
- identity: {}
```

3. Master-3에서 kube-apiserver 컨테이너를 다시 시작합니다.

```
cloud-user@smf-data-master-3:~$ sudo docker ps -f "name=k8s_kube-apiserver" -q | xargs sudo  
docker restart
```

사후 확인

마스터에서 Kubernetes 확인:

```
cloud-user@pod-name-smf-master-1:~$ kubectl get pods -A -o wide | grep kube-apiserver
```

이제 모든 Pod가 작동해야 하며 다시 시작하지 않고 실행해야 합니다.